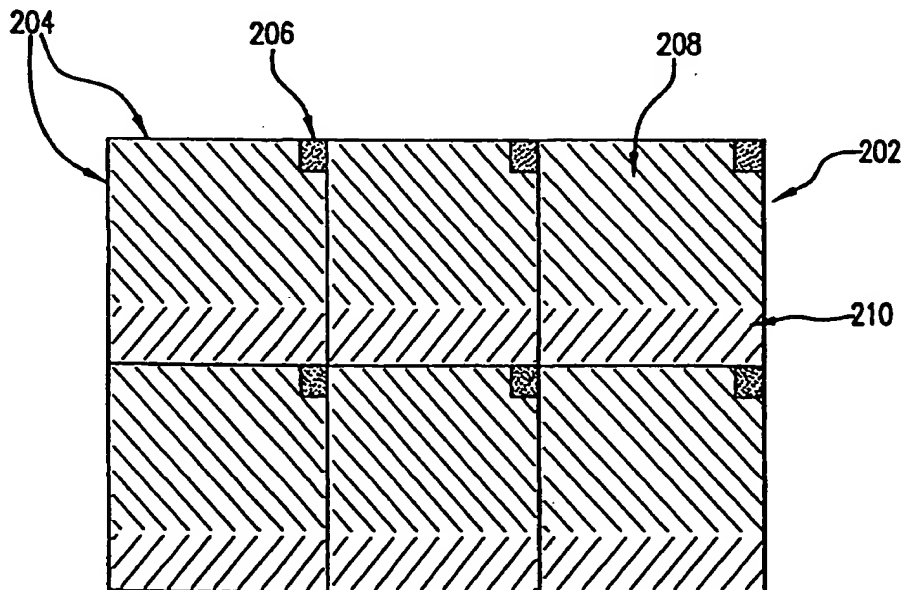




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06K 19/06	A1	(11) International Publication Number: WO 99/50787 (43) International Publication Date: 7 October 1999 (07.10.99)
(21) International Application Number: PCT/US98/20597 (22) International Filing Date: 30 September 1998 (30.09.98) (30) Priority Data: 9806973.5 1 April 1998 (01.04.98) GB 9806977.6 1 April 1998 (01.04.98) GB 9807001.4 1 April 1998 (01.04.98) GB (71) Applicant (for all designated States except US): XEROX CORPORATION [US/US]; Xerox Square 020, Rochester, NY 14644 (US). (72) Inventors; and (75) Inventors/Applicants (for US only): DYMETMAN, Marc [FR/FR]; 3, rue Pierre Duclot, F-38000 Grenoble (FR). COPPERMAN, Max [US/US]; 233 Sunset Avenue, Santa Cruz, CA 95060 (US). (74) Agents: OLIFF, James, A. et al.; Oliff & Berridge, plc, P.O. Box 19928, Alexandria, VA 22320 (US).		(81) Designated States: JP, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i>

(54) Title: CROSS-NETWORK FUNCTIONS VIA LINKED HARDCOPY AND ELECTRONIC DOCUMENTS



(57) Abstract

An identifiable sheet includes an area of sheet medium and, within the area, machine-readable markings that define two or more zones (202). The markings within the zones indicate a sheet identifier (208) identifying the identifiable sheet. The orientation of the sheet is given by an orientation marker (206). The position of the zones are given by markings (210) representing the zones position.

BEST AVAILABLE COPY

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Cross-Network Functions Via Linked Hardcopy And Electronic Documents

The present invention relates to functionalities of computers and networks of computers, and more particularly to the provision of cross-network functions via linked hardcopy and (digital) electronic documents.

It is well known to provide for the linking of electronic documents by setting up hyperlinks between documents in HTML format stored at, and transferable between, computers forming nodes of a conventional computer network, e.g. internet, intranet. See Berners-Lee T.J., Cailliau R., & Groff J.-F., *The Worldwide Web, Computer Networks and ISDN Systems* 25 (1992), pp. 454-459. North-Holland.

There has also been made available technology (so-called SmartPaper; Xerox Corp.) whereby documents sent by fax can contain, in addition to the human-readable information (e.g. text), printed machine-readable encoded data. Once the fax document is received the encoded data is decoded at the receiving fax device, and an appropriate additional function is performed by the receiving device. Such functions may include the sending of a specified document to the sending fax device.

However, the above techniques have considerable limitations in terms of accessing information (e.g. multimedia information) related to a hardcopy document in the user's possession at a given time. The first relies entirely on conventional UI techniques — keying in and/or point and click — with some personal networked computing device; and this requires the user to divert his attention away from the hardcopy document (e.g. a book) which he has been reading and it requires the user (a) to manually enter information (e.g. a WWW URL) needed to retrieve the related information and (b) if he doesn't know beforehand and has not been informed in the hardcopy document where to access the related information, to perform some search in order to find it.

The second technique provides only for a limited set of information of a limited type (printed documents) to be retrieved and viewed. Again, the user is required to stop reading the document, to mark it in appropriate way, and then to feed it manually into a fax machine or scanner coupled to a networked computing device, in order for the desired function to be performed.

Substitute Sheet (Rule 26)

There is a need for techniques which provide in a non-disruptive streamlined manner related information and related functions via a user's interactions with a hardcopy document.

The present invention provides a method carried out in a printing system, for producing a plurality of imageable substrates, comprising: (a) providing a plurality of sheet media; (b) for each sheet medium, printing coded machine-readable markings on the sheet medium to produce said substrate, said markings being printed so as to define a plurality of zones on the substrate, each zone containing machine readable first markings defining a page identification code, the page identification code being unique for each substrate.

Preferably, each zone further contains machine readable second markings defining a page location code, the page location code being unique for each zone and defining the position of the zone on the surface of the substrate.

The invention further provides an imageable substrate, comprising: a sheet medium; coded machine-readable markings formed on the sheet medium, said markings being formed so as to define a plurality of zones on the substrate, each zone containing machine readable first markings defining a page identification code, the page identification code being unique to the substrate.

The invention further provides a method of printing a document, each page of the document being printed on a respective substrate, each substrate being in accordance with any of claims 6 to 10 of the appended claims, comprising, for each page of the document: (e) deriving print data, said print data defining human-readable information to be printed on the page, said human readable information comprising at least one active element, (f) generating first association data, said first association data defining a mapping between the or each active element and the page location code within one or more corresponding zones of the substrate, said corresponding zones being overlapped by or adjacent a respective active element when the active element has been printed, (g) storing said first association data, and (h) using said print data, printing said human readable information on the substrate.

The invention further provides a printed document, each page of the document being printed on a respective substrate, each substrate being in accordance with any of claims 6 to 10 of the appended claims, the document further including human readable information printed on the substrate.

Substitute Sheet (Rule 26)

Preferably, said human readable information comprises at least one active element, the or each active element and the page location codes within one or more corresponding zones of the substrate being associated, said corresponding zones being overlapped by or adjacent a respective active element in the printed document.

5 The invention further provides a method carried out in an information processing system having a processing device coupled to a network and a handheld manually operable and positionable graphical data input device and an information output device coupled to the processing device, comprising: (k) in response to the user positioning the graphical data input device for image capture at a location in a
10 printed document according to claim 16 or 17 and a first user input being entered by the user via the graphical data input device, deriving first image data defining a two-dimensional first image, (l) using the first image data, extracting the page identification code from one or more zones of the substrate present with said first image, (m) using the page identification code obtained in step (l), generating
15 instruction data, said instruction data including said page identification code and a request for retrieval of a corresponding electronic document, (n) transmitting said instruction data over the network, (o) receiving over the network an electronic document corresponding to said page identification code, and (p) outputting said electronic document using said information output device.

20 Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

 Figure 1 illustrates schematically the scenario, provided by the invention, whereby a user may access electronic documents via interaction with hardcopy documents;

25 Figure 2 shows schematically the routing of various data, including page identification codes and page location codes, over a network;

 Figure 3 illustrates a sample of zones, and the disposition of machine readable data, on an imageable substrate in accordance with one embodiment of the invention;

30 Figure 4 is an illustration of an example of a printed document showing its components in accordance with the invention;

 Figure 5 shows how digital information is encoded in the machine readable data illustrated in Fig. 3;

Substitute Sheet (Rule 26)

Figure 6 shows the disposition of zones on a page, and an exemplary zone, in accordance with one embodiment of the invention;

Figure 7 illustrates an exemplary zone in which machine readable data is encoded, in another embodiment of the invention;

5 Figure 8 is a schematic illustration of the internal components of a graphical data input device or pointer in accordance with one embodiment of the invention;

10 Figure 9 shows a configuration for passing page identification codes and page location codes to the network using a graphical data input device or pointer in accordance with another embodiment of the invention;

Figure 10 illustrates the capture of an image containing coded zones using a graphical data input device or pointer in accordance with one embodiment of the invention;

15 Figure 11 shows a variant of the graphical data input device or pointer in accordance with one embodiment of the invention, i.e. integral with a writing implement;

Figure 12 is a flow chart of the processing steps performed by the local devices in implementing an embodiment of the invention;

20 Figure 13 is a flow chart of the processing steps performed by the remote device (at which the digital electronic document is stored) in implementing an embodiment of the invention; and

Figures 14 and 15 illustrate isomorphism between physical (hardcopy) and electronic documents.

25 Reference is made herein to 'electronic document'. It will be appreciated that this is intended to cover not only electronically stored files which can be presented to the user visibly (e.g. via printing or electronic display of a page or document), but also to audio and audio-visual documents and any other records of information that can be stored and retrieved electronically and presented by appropriate means to the user.

30

1. System overview

1.1 Basic architecture

Figure 1 illustrates schematically the scenario, provided by the invention, whereby a user may access electronic documents, and cause functions to be performed at a local or remote (networked) device, via interaction with hardcopy documents.

The user may manipulate (e.g. position or move with a 'wiping' action) a special pointer 502 (described in detail hereinbelow) to capture local images of portions of a document 2 and feed data derived from the captured images to the network (e.g. via some local device (not shown) provided with network communications facilities): the elements of this data are discussed in further detail hereinbelow. The document 2 is, or is formed from, a special substrate (discussed in further detail below with reference to Figs 4 to 7), hereinafter referred to as "coded substrates", in which visible or invisible machine readable markings defining a page identification code and, optionally, a page location code are formed.

Using this infrastructure, certain functions, e.g. the presentation to the user via a local device (peripheral) 4 of a digital page related to the document 2, or an operation upon information related to or present in the document 2, can be performed.

Figure 2 shows schematically the routing of various data, including page identification codes and page location codes, over a network between the main architectural elements of the system.

The user can click a button (not shown) of the pointer (similar to a mouse click) either at a certain location on the physical page 2, or on a small sticker 2' (similarly formed from a coded substrate and shown at the bottom left), on the peripheral 4.

The pointer 502 preferably decodes two pieces of data — the page-identifier (pid) and the localisation on the page (loc). These decoding actions are shown as dotted lines.

The page-identifier pid (or pid') is a number which needs to be resolved into a network address. This is done by sending it to a router 802, whose address is known *a priori* by the pointer 502. The router 802 returns two pieces of information: a network address for the page-identifier, and a type which can be either "peripheral"

or "digital-page". The association of pid (or pid') with its address and type is kept by the pointer in a cache for possible future references.

When the type of the page-identifier is "peripheral", the pointer 502 assigns the network address of the peripheral 4 to its peripheral-address store. The localisation loc' is not used. When it is "digital-page", it (1) assigns the network address of the digital page 6 to its digital-page-address store, and (2), it sends the page localisation loc and the peripheral address to the digital page address. This is discussed further below with reference to Figure 12. The digital page 6 may be a digital (displayable/printable) representation (e.g. Web page) of the printed (human-readable) information in the document 2, or may be a digital representation of some document or information related to, or derivable from, the printed document 2.

When the digital page (located on a server (not shown), e.g. a conventional Web server on the network) receives data from the pointer 502, it first decodes the localisation loc into some action which typically produces output, and then sends this output to the peripheral address. The digital page then waits for new loc and peripheral-address data. This is discussed further below with reference to Figure 13.

1.2 Further network considerations

1.2.1 *page-id* and network routing

Addresses can be interpreted globally (similar to Internet addresses) or "pre-empted" locally by home computer

A variant is for an entity producing coded substrates ("coded substrate supplier") to associate a default Web page, at some centralised site, with any *page-id* number it produces. The buyer of a coded substrate page *ipso facto* becomes the owner of this centralised Web page. He may have only limited rights as to using this page (for memory usage considerations), as for instance only the right to memorise in it the net address of his own Web page, which contains the detailed data. The centralised Web page provided by coded substrate supplier acts only as a "go-between" between the *page-id* and its associated digital page. The interest of this scheme is that from anywhere in the world, the *page-id* can be routed to its corresponding centralised Web page through a generic routing scheme.

Of course, optimisations of this scheme should be found, which do not require the owner of an address book living in Paris to route each *page-id* first through the headquarters of the coded substrate supplier in Palo Alto then back to his home computer in Paris.

5 Another optional feature is caching network path associated to a *page-id* for access efficiency.

1.2.2 Wireless communication, ubiquitous peripherals

Functions that the communication infrastructure may provide include:

- 10 • allowing the pointer to send the *page-id* code to a predefined address router, which is able to determine the Internet address of the digital page corresponding to this *page-id* ,
- getting back (and possibly caching for future use) this page Internet address,
- (possibly) sending a *page-id-code* to the page Internet address,
- 15 • sending *pointer-loc* codes to the page Internet address, .
- sending to the page Internet address the Internet address (or phone number, etc.) of the peripheral to which output should be sent,
- sending back to this peripheral the output produced by the *pointer-loc* interpretation process associated to the digital page.

2. Encoding data on substrates

20 Current or shortly available printing technology, conservatively estimated, permits to print on the order of 1000 bits of reliable information per square centimetre (according to recent public releases, the Glyph™ technology developed by Xerox

25 currently permits to encode on the order of 500 bits/cm2, including a robust error correction scheme, and assuming printing and scanning at 400 dpi. Considering the special constraints placed on Glyph™ (unobstructiveness to human visual perception), which are not shared by our invisible marks, and also keeping in mind that the production of coded substrate will be a process carried out by specialised

30 machines, our assumption of 1000 bits/cm2 seems reasonable).

A recent CACM article about address protocols currently being devised for the Internet notes that 128 bits theoretically permits to address 3.4×10^{38} items, or 6.7×10^{23} items per square meter of the surface of the Earth. Because in reality

Substitute Sheet (Rule 26)

"the assignment and routing of addresses requires the creation of hierarchies that reduces the efficiency of the usage of the address space", these figures should be lowered somewhat in practice, a pessimistic estimate being 8×10^{17} nodes (1564 addresses per square meter of the Earth), an optimistic one 2×10^{33} nodes (4×10^{18} addresses per square meter of the Earth).

It can be safely assumed that the data zone in each cell holds 256 bits of reliable information. Of these, 128 bits hold the page-id (which is thus redundantly repeated on each cell of the page), 32 bits hold the cell localisation (or cell address) on the page (this corresponds to a maximum number of 65536 cells on a page, this corresponding to a 123cm x 123cm maximum page). 224 bits are left for the page-id-code (and possibly other information, such as publisher's private data).

We will now assume that the pointer device has the shape of a pen, and that its tip, when positioned anywhere in a given cell is able to "see" the whole cell. By decoding the cell address, the pointer knows the localisation of the cell on the page. By seeing the cell border, the pointer is also able to determine its precise position inside the cell. Thus the pointer is able to localise itself precisely relative to the whole page. The pointer is also able to read the page-id and the page-id-code and therefore to send these two, along with its page localisation, for interpretation via the external infrastructure.

The code scheme we have just described is but one of several possibilities.

2.1 Invisible code marking

Let us now turn to another point: how does the coded substrate supplier produce the just described codes as invisible marks on paper, and how these marks can then be recovered by the pointer device.

One option is the use of ultra-violet ink to print the invisible codes on white paper. Such an ink looks transparent to the human eye, yet an area of paper covered by it reflect UV light differently from an uncovered area.

In this case, the coded substrate supplier produces sheet of paper in different formats for uses by the publishing industry. Each sheet is processed through a specialised printing procedure which (1) assigns a fresh page-id (and possibly page-id-code) to this sheet, and (2) prints in UV ink the corresponding data cells on the surface of the sheet.

Substitute Sheet (Rule 26)

The publisher buys these apparently uniformly white sheet and now prints, in standard ink, visible marks on them.

The end user positions the pointer device on the page. The pointer emits UV light and reads back the UV light reflected by the UV-ink layer (rather than emitting UV light, the pointer might also exploit the UV component of natural "white" light). This light, apart from being invisible to the human eye, is similar to conventional light and "standard" techniques for optical scanning can be applied to recover the UV-ink data.

In certain cases it may be necessary to use different inks for code marking to avoid problems of the conventional ink layers blocking the transmission of the UV light. However, in standard trichromatic printing, three layers of coloured inks (yellow, magenta, and cyan) are superposed to produce all visible colours (in fact, for practical reasons, quadrichromatic printing, where black is the fourth "colour", is more often used than trichromatic printing, but let's ignore this here). Each layer filters one part of the light spectrum independently of the others. That is, the magenta layer does not block light from the cyan and yellow layers, etc.

Other to UV ink are infrared ink, phosphorescent ink (phosphorescence is used in TV screens) or some type of magnetic ink (which has been proposed in the past for bar-code type scanning). It should also be remembered that it is not essential that the codes be encoded through optical means. The only crucial properties are that they be durable, do not interfere with the visible printed marks, and permit to recover page-id, page-id-code and pointer localisation.

Figure 3 illustrates a sample of zones, and the disposition of machine readable data, on an imageable substrate in accordance with one embodiment of the invention: this shows substrate markings consisting of four elements —

- cell borders, which must be distinguishable from cell contents
- some indication of the orientation of the page
- page-id*, represented within the cell border
- localisation information, represented within the cell border.

Each zone or cell 202 includes a border 204 and an orientation marker 206. A first set of markings 208 over part of the interior of the cell 202 are encoded representations of the page-id, while a second set of markings 210 over a (smaller) part of the interior of the cell 202 are encoded representations of the localisation

Substitute Sheet (Rule 26)

(page-loc) - uniquely defining the position of the cell 202 within the page. The orientation of the page is given by a mark in one corner of the cell, making it asymmetric in both dimensions. The cells are preferably tiled to cover the entire page.

In Fig. 3, the page-id and localisation are shown schematically; detail is given in Fig. 5, discussed further hereinafter.

Figure 4 illustrates the components of a printed document as printed on a coded substrate. The printed document 102 comprises a layer 104 of printed visible (human-readable) information (document content) printed on a coded substrate 106. The coded substrate 106 in turn comprises a layer 108 of visible or invisible machine readable markings (encoding the page-id and localisation), printed on a sheet medium 110 (e.g. paper).

Figures 5A and shows how digital information is encoded in the machine readable data illustrated in Fig. 3, i.e. the cell or zone contents. Fig. 3 shows the binary data, i.e. 47 bits of page-id in the upper section 302 (the bit stream wraps at the cell border 204), and 16 bits of page localisation data (loc) in the lower section 304). The page-id code denotes 108429159095492 = 629DA182DCC4 (hexadecimal) = 110001010011101101000011000001011011100110001000001000000010101 (binary; to make the wrapping explicit). Within the border of a cell, the page-id is represented as an ordinal number, in binary (Fig. 5A). The cell is 8 by 8 symbols (or bits) wide, with one symbol's worth of space taken up by the orientation mark. In the 16 bit loc code in section 304, there are 8 bits for the X co-ordinate and 8 bits for the Y co-ordinate. Thus, for the cell (zone) shown, its position is 16,21 on the substrate. Localisation identifies a cell (zone) on the page, not a point on the page.

With this encoding scheme, a complete cell must be within the region of the page from which page-id and localisation are to be retrieved. Less than a complete cell does not contain sufficient information.

The addition of error correction information is desirable. This can be done with standard techniques and is not shown; the actual encoding after the addition of error correction information will be different from what is shown, and will require more bits (and thus more space).

The number of possible pages and the number of possible cells on a page are a function of the density of the encoding. Using the encoding shown, 2 to the 47th pages can be identified, with up to 64 by 64 cells on each.

Substitute Sheet (Rule 26)

Fig. 5B shows the same data as in Fig. 5A, but represented by Data Glyph markings. Encoding using data glyphs and the retrieval of data therefrom is discussed further in US-A-5,486,686, EP-A-469864, and the abovementioned GB application (ref.R/98003/JDR). Here, there is a first set of glyphs (markings) in upper section 402 and a second set in lower section 404, the two sets of glyphs being encoded representations of page-id and loc codes. DataGlyphs offer advantages in robust decoding.

In accordance with the invention, the page-id, page-loc and any other data can be encoded on the coded substrate in any other suitable manner. For example, the encoding may be as described in any of: U.S patents 4,786,940; 5,245,165; 5,278,400; 5,315,098; 5,449,895; 5,521,372; 5,537,223 ; 5,572,010 and 5,576,532.

The inks used to encode the abovementioned data may be visible or invisible, and may be detected by visible optical, IR, UV, magnetic or any other suitable means. For example, suitable inks for the markings encoding the data may be as described in: U.S patents: 5,075,186; 5,225,900; 5,301,044; 5,145,518; 5,046,087; 5,275,647; 5,208,630; 5,202,265; 5,271,764; 5,256,193; 5,286,286; 5,331,140; or 5,291,243.

Figure 6 shows the disposition of zones on a page, and an exemplary zone, in accordance with one embodiment of the invention.

Figure 7 illustrates an exemplary zone in which machine readable data is encoded, in another embodiment of the invention. Here, a paper page is divided into 0.5 cm x 0.5 cm square cells. Each such cell can contain on the order of 256 bits of information. The cell is identifiable by its distinctive border, consisting of uninterrupted horizontal and vertical lines of black pixels. A square of pixels, which is separated from the black border by a white area one pixel wide, holds the data (in the illustration this square is shown 10 pixels x 10 pixels wide, but in reality it would have a higher resolution).

3. User-positionable pointer/scanner

This device may take the form of a pen-shaped pointer, provided with clicker. Additionally/ alternatively the pointer may include a small transparent screen, rimmed with a code-reading device, with a cross hair in the middle: this may be a less

costly form in the nearer-term than a pen-shaped pointer, and it can encompass more of the surface of the page at one time.

Figure 8 is a schematic illustration of the internal components of a graphical data input device or pointer in accordance with one embodiment of the invention. The pointer is a complex object that may be a single physical device or
5 may be several communicating devices. It contains

1. A hand-held image-reading device, containing a button.
2. An image-decoding device.
3. A network communication device.

10 The image-reading device receives light and produces image data. When the user clicks the button, the image data is made available to the image-decoding device. That device decodes the image, producing a <pid, loc> pair which is used by the network communication device as described in Fig. 12.

Figure 8 shows schematically a special purpose device (pointer); and the
15 elements that comprise the pointer are a camera 802, a frame grabber 804, memory 806, a CPU 808, a button (not shown; similar to a conventional mouse button) for the user to click on a region of a page (which activates the frame grabber 804), a memory 809 storing image decoding software, network communication software, and control software, and network connection hardware 810. (The functionalities of software can
20 be reproduced in hardware if that is advantageous.) Included in the memory 809 is storage for a digital-page network address and an output-peripheral network address and a cache comprising a set of triples <pid, type, address>, where "type" distinguishes the pid of a digital page from that of a peripheral.

The image-reading device can be realised as a pen-camera or hand-held
25 scanner (typically CCD elements with hardware to turn optical input into image data).

The pointer can also be realised as a collection of communicating devices. The most obvious separation points are mentioned in 1, 2, and 3 above, though 1 and 2 could be in the same physical device, or 2 and 3.

The image decoding device may also be realised as a general purpose
30 computer containing a frame grabber board and an image decoding software package. Likewise the network communication device can be realised as a general purpose computer with a network connection or as a network computer.

Substitute Sheet (Rule 26)

The function of a frame grabber is to capture a snapshot in memory from the image data that the camera continuously produces. If the camera is not integrated with or tethered to the image decoding device, both devices must have wireless communication facilities to allow the camera to transit the image data to the frame grabber. (There are performance considerations, because image data is large.) The button click also must be communicated, as it resides on the camera for ergonomic reasons, but activates the frame grabber.

The snapshot is input to the image decoding device, whose output is the <pid, loc> pair. If the image-decoding device is not integrated with the network communication device, the <pid, loc> pair must be transmitted to the network communication device. Again, this requires each device to have communication facilities (a cord, or some form of wireless communication such as infra-red, or mobile telephony).

An alternate description of the pointer is shown in Fig. 9: this shows a configuration for passing page identification codes and/or page location codes from the pointer of 11 to a network computer, in accordance with an embodiment of the invention. The image capture device (e.g. CCD camera) 506 is coupled by wired or wireless (e.g. IR or RF) link to processing device 602 and in use provides image data defining capture images to the processing device 602. The operative elements of the processing device 602 are a frame grabber circuit 604, image decoding software 606, and a CPU 608, which are known in the art. (In certain embodiments, the camera 506 and processing device 602 may be combined into an integral handheld unit; Fig. 8). In use, the processing device 602 extracts from the image data the corresponding page-id and page-location data (<pid, loc>) and communicates them in a wired or wireless fashion to a local device (here, a network computer 610, which is linked to the network (intranet, internet) in a known manner). The computer 610 has its own unique network address, but need not have any information output device (e.g. display screen, printer).

Here it is the network computer that has the cache and the digital-page network address and output-peripheral network address storage. As the network computer need not have a screen, it does not have to be a large device.

Figure 10 illustrates the capture of an image containing coded zones 202 using a graphical data input device or pointer in accordance with one embodiment of

Substitute Sheet (Rule 26)

the invention. There is shown a small region of a page, with the field of view 702 of the camera (not shown) delimited by the circle. The radius r of the camera's field of view must be larger than the diagonal of a cell 202 in order that the field of view contain at least one cell 202. The centre of the camera's field of view (shown via cross-hairs 706) will lie in exactly one cell. (The centre 704 may lie on a cell border. In that case, one of the adjacent cells is chosen by convention.)

The location of the centre 704 relative to the cell borders identifies a location within the cell 202, which in combination with the orientation and the identification of which cell 202 it identifies the location on the page.

Figure 11 shows a variant of the graphical data input device or pointer in accordance with one embodiment of the invention, i.e. integral with a writing implement. The pointer 502 comprises a marking device 504 (which may be a pen or any other marking device suitable for making marks which are visible to a user), and an image capture device 506. In use, whether or not the user is making marks using the marking device 504, the image capture device 506 is able to capture images of an area A of a document 508. (For the sake of illustration, the sizes of these elements are exaggerated - e.g. in practice, the area A may be much closer to the tip 505 of the marking device 504 than appears). In certain embodiments, the marking device 504 may be omitted.

The document 508 may be a 'blank' coded substrate, or such a substrate having human-readable information printed thereon.

4. Processing captured information

Figure 12 is a flow chart of the processing steps performed by the local devices in implementing an embodiment of the invention.

Figure 13 is a flow chart of the processing steps performed by the remote device (at which the digital electronic document is stored) in implementing an embodiment of the invention.

5. Variants and extensions

5.1 Possible Simplifications

This sketch of coded substrates, documents produced therefrom and applications is provided in a quite general form, which requires in certain

Substitute Sheet (Rule 26)

embodiments a communication protocol and infrastructure, including the need for widely available wireless communication between pointers, peripherals, and computers.

These requirements may be relaxed at the cost of a tighter coupling of coded substrates to peripherals. For example:

- Shakespeare's play Othello could be published, with similar capabilities, as a book/CD/pointer/software package for use with a computer containing a CD player. If it is assumed that a CD player that can interact with a monitor, it could be published as a book/CD/pointer package, where the software is built into the pointer (which controls the CD player).

- If it is assumed that wireless telephony is available and the pointer can contain a telephone, an coded substrate sales catalogue could allow a user (called, say, Ariane) to place an order by clicking on an item. The catalogue provider's phone number could be encoded in the substrate along with the page-id and pointer location information. Clicking on an item then initiates a call to the encoded number. The page-id and pointer location are thereby transmitted to the provider. These identify both the item and the customer, allowing order to be placed.

5.2 Extensions

5.2.1 Extensions of the pointer device:

- Sensitive transparent screen overlaid on a coded substrate. Here, a flexible, transparent screen made of plastic (or rubber, etc.) is provided, in one (or several) of its corners, with a flat coded substrate code reading device, and which, when touched with a finger, is able to determine the location of the finger relative to itself. Such a screen can for instance serve as a universal keyboard: overlay it over a coded substrate on which a keyboard has been drawn. The user "types" at a certain location, and the screen "knows" what this location is, not only relative to its own co-ordinates, but also (because of the information read at the corners) relative to the intrinsic paper co-ordinates. The infrastructure can then determine that this is the location of letter A.

Such transparent screens could be more practical than a pointer when several fingers act in parallel, like when typing on a keyboard. This also makes possible implementing some "paper-based" musical instruments on this principle.

Substitute Sheet (Rule 26)

[With more sophisticated technology, it might be possible to provide the transparent screen with a display capacity. This could be used to overlay the printed marks on physical paper, seen by transparency, with dynamically displayed graphics, in a way sensitive to the location of the screen on the page.]

5 Clipboard variant. A variant of the transparent sensitive screen idea is to have fingers touching the paper directly, while finger locations are recorded through a "clipboard" positioned under the paper, rather than over it.

10 Pointer Gloves. Another device which would not require either a screen or a clipboard, but which would have the same functionality, would be gloves with finger tips holding small pointer like reading devices.

- Bi-directional writer-pointer and hand-held "stamper-like" printing machine.

15 A variant of the writer-pointer (Fig. 11) is that of a "bi-directional" writer-pointer, that is, of a device which receives external commands to deliver or not to deliver ink depending on its current position relative to the paper sheet. The user covers an area by quick forward and backward moves of the pen; the positions at which ink has been deposited approximate a piece of graphics sent by the external site; in this way a limited "local printing" capability can be simulated.

20 A perhaps better way of obtaining a local printing capability is through a small hand-held printing device, resembling a "stamper". This device can be placed by the user over areas of the paper sheet; for each such placement, the stamper is able to determine its precise position relative to the sheet co-ordinates, and it also "knows" which parts of the area still remain to be covered by marks. By repeated rough placements of the device by the user, a piece of graphics can be printed with high
25 precision at a specific location on the paper sheet.

- Hand-held precision scanner-pen

30 The precise localisation capability rendered possible by the coded substrate can be exploited for scanning purposes. Suppose the user has produced some drawing or note in the margin of a coded substrate. A small pen-like device can be provided both with a localisation capability and with a scanning capability. By moving this device over the area of interest, if necessary through repeated moves, the user can perform a precise digitalisation of the whole drawing; this can be done even

Substitute Sheet (Rule 26)

if the device has a small "footstep" (active scanning surface), because the images scanned in over several moves will be easy to connect into a global picture.

5.2.2 Extensions of the paper substrate:

5 • Coded Confetti. These are small round stickers made from a coded substrate. In particular, each sticker has a unique page-id. They can be used to add multimedia links to a piece of conventional paper. The user buys a few of these coded confetti. He is perusing a conventional document, printed on standard paper, and wants to associate a certain action to a certain location on a given page of this
10 document (this action could be to go to a certain Web site, to ring Mary's phone number, to print an address on an envelope). He sticks the confetti (which could be coloured or transparent) at the location of interest and associates (through his personal computer, or through a service provided by the confetti provider) this action with the confetti's page-id.

15 Coded Confetti can be seen as a transition technology that permits to confer to conventional documents some of the advantages of CODED SUBSTRATE documents. Obviously these confetti could be stuck on any support, not only printed material. A confetti stuck to a machine part could have the effect of showing a description of this piece on a screen, when pointed to.

5.2.3 Extensions of the coded substrate document architecture:

20 • Interdocument communication. In the embodiments previously discussed, it has generally been assumed that every desirable capability of a coded substrate document is provided by its publisher independently of the existence of
25 other coded substrate documents. Here, it is briefly described how communication can be established among coded substrate documents produced by different publishers, increasing overall document functionality, on the basis of two examples.

30 • Computerless email: Ariane wants to send a message to her friend Irene. She writes the message, using a writer-pointer (Fig. 11), on NotePaper, a pad of coded substrates designed for sending notes. She then looks up Irene in her address book, itself an coded substrate document. She clicks the pointer on a Print icon in Irene's listing. She then clicks on a Send icon on the sheet of NotePaper containing the message. The NotePaper application sends the image of that sheet of NotePaper to

Substitute Sheet (Rule 26)

the address book application, which locates Irene's fax number in its database, and faxes the message to Irene.

• Comprehension Aid: Ariane is at home reading an English novel in a coded substrate edition that does not provide translation. She decides to get a French gloss of a difficult paragraph. From her wallet she takes a "utility" card with a number of application icons, and clicks on English->French Glosser. She then clicks on Send Region on the inside of the back cover of her novel, and subsequently on the beginning and end of the difficult paragraph. The novel's application sends the paragraph to the glosser application, which displays the paragraph with a French gloss on Ariane's television screen.

6. Additional functionalities

6.1 Basics

Terminology

A digital page is an object comprising functions and data that is constructed as the active counterpart to a physical page. As such, it implements any actions initiated by the user interaction with the physical page. The term "digital page process" refers to an executing instance of a digital page.

6.1.1 Communication between the Physical Page and the Digital Page

The user interacts with a coded substrate page by positioning the pointer on the page and clicking a button on the pointer. The pointer recovers its position relative to the page frame.

The coded substrate communication infrastructure, described elsewhere, delivers the position and the clicking events from the pointer to the digital page, delivers output from the digital page to a peripheral whose address has been specified, and delivers mode-switching commands from the digital page to the pointer. Communication between the physical page and digital page is limited to these kinds of information.

A digital page could do arbitrary computation with arbitrary data. What follows should not be viewed as a restriction on their capabilities. However, there are conventional capabilities that make sense for a digital page, that will be useful across

Substitute Sheet (Rule 26)

a wide range of coded substrate documents, and for which support can be provided in a generic manner; several of these are described here.

To reiterate, a digital page receives positions (x, y co-ordinates) and produces output (which may take the form of instructions to run a program on another machine, and input data for that program). The manner in which the co-ordinates are delivered to the digital page is not of consequence here, nor is the manner that the output is delivered to the machine on which it is to be displayed or executed; those elements are described elsewhere.

A basic coded substrate capability is to initiate an action by clicking in an active region on the page (which is typically graphically marked, as with icons, labelled boxes, or highlighted words). Active regions have two elements: regions and actions.

6.1.2 Regions

A digital page receives the position of the pointer, and must determine if that position is within an active region. To this end, it must have a description of the active regions on the page.

Rectangular regions have a particularly simple description. Given two diagonally opposing corners of a rectangle, you can tell whether a given point lies within it or not, and the function used to tell is also particularly simple.

Arbitrarily complex regions can be described in such a manner that you can tell whether a given point lies within it. The function needed to decide may be complex, and different types of descriptions require different functions. We will call a pair consisting of a description of a region and a function to decide whether a point lies in it a "region".

6.1.3 Actions

Once the digital page has determined that a position is within an active region, it must cause an action (or a number of actions) to be executed. It therefore must have a description of the action(s) for that region. Possible ways to describe actions include (but aren't limited to) programs (or functions), with or without input, pointers to programs (internal or external to the digital page), with or without input, and type, data pairs, where the type element identifies a conventional action to take, and the

Substitute Sheet (Rule 26)

data element is input (e.g., URL, <http://www.xerox.com> could mean "open the web page <http://www.xerox.com> in a browser").

6.1.4 Action Tables

5 An action table is a table of region, action pairs, where the action element of the pair may describe a sequence of actions. A digital page has at least one action table, describing active regions on the page. At any time, one action table is active, which is to say that positions delivered to the digital page will be looked up in that table.

6.1.5 The Output Player

10 A 'coded substrate-based action' (one arising through some user interaction with a coded substrate document using a pointer) often results in output on a machine near the user, while the digital page resides at some arbitrary location on the network.

15 This is why the output of a digital page may be input to a program to be run on another machine. It happens that in our examples, Ariane's local peripheral is her TV screen. In this case, output takes the form of a video signal sent to her TV. But if her peripheral were a computer, the output would need to be of a different form, and in fact, of a different form for each different computer platform. The digital page should

20 not be encumbered with the details of the form of the output. We assume that each peripheral is equipped with an output "player", so called because it knows how to play various types of data; sound data is played on the audio system, video is played with a video player, text is displayed in a conventional way on the screen. (Current browsers provide such capabilities in a platform-independent way for computers.)

25 We assume either that the peripheral can run an output player directly or that the peripheral is controlled by a computer that runs an output player which sends output of the right form to the peripheral.

 An output player receives typed data (video, audio, and text are types that have already been mentioned; others could include control information (volume,

30 colour, ...), spreadsheet position and value, what have you).

 In addition to giving the digital page platform independence, the output player solves the problem of how to push information and actions to another machine. A program on one machine does not routinely have permission to display data or

Substitute Sheet (Rule 26)

execute a program on another machine. In the case of coded substrates, the peripheral near the user is registered with the communication infrastructure. Either as part of the registration process or explicitly by the user at the point of interacting with a coded substrate document, an output player is launched on the peripheral. The digital page
5 can send data to the output player, whose address is known to it by virtue of the registration; it need not launch a program on the peripheral. The output player, in contrast, can launch programs on the peripheral (subject to local permissions).

6.1.6 Communication with Applications

10 In the course of executing actions, a digital page may launch other applications. When a digital page launches an application with which it may have continuing interaction, it maintains a connection with the application (such as a socket). The digital page may launch an application (for example, a browser) on the user's peripheral via the output player. The output player returns to the digital page
15 the information it needs (such as a socket name) to establish a direct connection with the application.

A digital page may also provide a mechanism to allow the user to establish a connection between itself and an already-running application.

20 6.2 Output Actions

Here we describe one way to implement the output actions attributed to user (hereafter called Ariane) herein. A variety of other possibilities exist.

6.2.1 Action 1: Text Output

25 Ariane is in doubt about the meaning of a word. She clicks on it; an explanatory note for the word is displayed on her TV screen.

The digital page is responsible for producing the explanatory note; how it gets to Ariane's TV screen is the responsibility of the communication infrastructure.

The active action table contains a pair W, E, where W is the region containing
30 the word, and E is, for example Output, Exp, where Output is a type meaning "display the data element of the action pair to the user" and Exp is the text of the explanation.

Substitute Sheet (Rule 26)

The digital page receives the position of Ariane's pointer. It looks up the position in the active action table and determines that it is in region W. It therefore executes action E, that is, outputs Exp, which is sent to the output player of Ariane's TV.

5 Note that this mechanism requires that the explanatory note be provided by the publisher of the document *a priori*, and presumably that the word be graphically identified as one that has an explanation. The ability to look up a word in an on-line dictionary would provide a definition for any word, without putting a burden on the publisher. [Of course, the dictionary definition may not be identical to the hand-

10 crafted explanatory note, which could take advantage of context. But in a great many cases, a dictionary definition would be valuable.] In order to provide this, we'll need a general text-selection mechanism. We'll return to this example in Section 6.5: Selecting Content.

15 6.2.2 Action 2: Audio Output; Action Table Switching

Ariane clicks on the RECITE box, then on the beginning of a line; the line is recited by Laurence Olivier over her TV speakers. The active action table contains an entry for the RECITE box. Its action is of a different type; it affects the state of the digital page process. It makes the active action table inactive, and makes some other

20 action table active—one that contains an entry for each line, as follows. The region element of an entry describes the region containing the line. The action entry has two parts. The first part says to output audio data (stored as part of the action), which is the recitation of the line by a famous actor. The second part restores the previous action table as the active action table. (Note that this mechanism of action tables, and

25 changing active action tables, is only one implementation of many. Another general mechanism would be to associate operators and operands with regions, distinguishing between the types of objects. Action 1, Text Output, would bundle the necessary operand with the operator, so only one click is necessary. For Action 2, one click would designate the reciting operation and a second click would designate the line as

30 its operand.)

This also could be done via text selection as opposed to *a priori* definition of the regions and provision of the recitations. It would require a reading of the play aligned to the text and segmented into lines; then selection of arbitrary text could

Substitute Sheet (Rule 26)

trigger location of the point in the reading, the audio data from there to the end of the segment could be output.

6.2.3 Action 3: Video Output

5 Ariane clicks on the PLAY SCENE box, then on the beginning of a scene, and a video of Orson Welles' interpretation of the scene is played on her TV.

This is just like the previous case, but the action table for the PLAY SCENE box has entries appropriate to playing videos of scenes.

10 6.3 Annotating; Adding Actions

The example is the following: Ariane has previously connected to the Web pages provided with the coded substrate document that she's reading (Othello), by clicking on the MAIN WEB PAGE box. As a result, she has a browser on her screen, open to one of these Web pages. She uses a normal pencil to make a note to herself
15 on the page. She clicks on the LINK TO WEB PAGE box. She clicks on the note she just wrote. Henceforth, clicking on her note causes the Web page to pop up on the screen.

A similar but slightly more general example is easier to illustrate, after which we'll return to the above. Consider: Ariane has a browser on her screen, open to an
20 arbitrary Web page. She uses a normal pencil to make a note to herself on the page. She clicks on the LINK TO WEB PAGE box. She clicks on the note she just wrote. A dialog box pops up on the screen. She copies the URL of the Web page from the browser to the dialog box and clicks "Apply" in the dialog box. Henceforth, clicking on her note causes the Web page to pop up on the screen.

25 The active action table contains an entry for the LINK TO WEB PAGE box. Its first action changes the mode of the digital page. Rather than continuing in its normal event-driven mode, the digital page waits for the next click from the user. When it's received, the position P of the pointer is recorded, rather than being interpreted as a standard interaction (it is not looked up in the active action table, as
30 would normally be the case). A region R of some conventional size with P at its centre is created. The above-described dialog box interaction is initiated, and the URL U given by the user is used as the data element of an action URL, U. The entry R, URL, U is added to the active action table. When the user subsequently clicks in

Substitute Sheet (Rule 26)

the defined region, the standard action-table mechanisms will pop the Web page up on the screen.

An ADD ACTION box would function in a very similar manner, the difference being that the digital page would not automatically interpret the data it gets from the dialog box as a URL. Instead, the user must provide a complete description of the action. For example, assuming actions are described as type, data pairs, the user could select the type from a menu and enter the data. A type "Program" that launches a program would provide a catch-all to allow virtually any type of action to be added.

Now let us return to an example. In this example, Ariane doesn't have to type in the URL because the digital page launched her browser, and thus can communicate directly with it. The browser provides its current URL when queried. [If browsers don't already have this functionality, they can certainly be extended to provide it.] Thus the dialog box interaction with the user is replaced with an interaction between the digital page and the browser, while the rest of the scenario remains the same.

6.4 Using the Pointer as a Mouse

The example is: Ariane clicks on the MOUSE MODE box and subsequently uses the pointer as a mouse, with the page as the mouse-pad.

Let's complete the example by saying that the MOUSE MODE box acts as a toggle; if she clicks on MOUSE MODE when in MOUSE MODE, the pointer returns to its non-MOUSE-MODE mode of operation. And let's be explicit about the context: some application is running, displayed on Ariane's TV (more generally, on the user's local peripheral). The digital page has a connection to this application, either explicitly established by the user or because the application was launched by the digital page.

The standard mode of the pointer is to initiate communication with the digital page each time the user clicks, sending its current position and the click event. Thus in non-MOUSE-MODE operation, the digital page gets the position of the pointer whenever the user clicks.

In MOUSE MODE, the digital page gets the position of the pointer at regular intervals, without the user clicking (and with no click event transmitted). When the user does click, the click event and position are sent as usual.

Substitute Sheet (Rule 26)

How It's Done. The active action table contains an entry for the MOUSE MODE box. Its actions follow: A command is sent to the pointer, to change its mode of operation. In the new mode, the pointer initiates communication every n timer ticks, sending the digital page its position each time, in addition to sending its position with a click event when the user clicks. [This change of mode, and thus the need for the pointer to receive control information from the digital page, can be avoided at the cost of more communication traffic, by having the pointer always send its position every n ticks. In non-MOUSE-MODE, the digital page would ignore positional information that is not accompanied by a click event. Alternatively, the pointer could distinguish itself whether it's in mouse-mode or non-mouse mode (e.g., the button is held down during mouse mode, or alternatively, positions are sent every n timer ticks when the pointer is moved in contact with the page).]

The mode of the digital page process is changed, too. The first position RP that the digital page receives with no accompanying mouse click is recorded. A cursor is displayed on the output screen, [cursor display is one of the capabilities of the output player] and its position CP on the screen is recorded. Each time a position P is received, the difference between P and RP is computed. That difference is added to CP and the cursor is displayed at the new value of CP. RP is replaced by P, and the process iterates.

When the digital page receives a position P with a mouse click event, it updates CP and RP as usual, and in addition it forwards the mouse click event, with CP substituted for P, to the application. By "forwards the mouse click event" we mean that the digital page puts the same type of mouse click event (changing the position) in the application's event stream.

Of course, if the application is the digital page itself, these issues are absent.

Absolute Positioning. What has just been described is how to use positional information relative to the printed page frame to simulate a standard mouse. As a positioning interface, this has the advantage of being familiar to mouse users.

This is relative positioning: the user moves the pointer relative to the current position of the cursor, and the absolute position of the pointer on the page is irrelevant.

Substitute Sheet (Rule 26)

An alternative is absolute positioning. Here the user moves the pointer relative to the page frame; the absolute position of the pointer is crucial. You want the cursor up near the top right-hand corner of the screen? Put the pointer near the top right-hand corner of the page.

5 Imagine that the page is the same size and shape as the screen. Then the absolute position of the pointer relative to the page frame could be directly translated to the screen; this combines elements of using a mouse and pointing at a touch-screen.

10 The printed page is rarely the same size and shape as the screen. However, the page dimensions can be scaled to match the screen (or to match some window on the screen), and the position on the page can be interpreted according to the scaled dimensions as a position on the screen (or in the window).

6.5 Selecting Content

15 We saw in Section 6.2.1 that in order to provide an explanatory note about a word, the note must be stored in the action table entry for the region around the word. The author must decide to provide a note for that word as part of the publication process. But suppose we want to support cut and paste from a coded substrate page into another document. We can't know ahead of time what chunks of text the user might want to cut. Suppose we want to help a foreign reader understand the

20 document, by allowing her to display a translation of some part of the text. We can't afford to translate the document into all languages. We'd like to offer machine translation as one option, but we can't know ahead of time what part of the text she'll need help with.

25 For a complete solution, we need what is referred to herein as *isomorphism* between the physical page and the digital page. But simple text selection can be done using action tables.

Let's take a simple method of selection to begin with. Suppose you select a word or an object by double clicking on it.

30 This can be supported with an action table with an entry for each word and each selectable object. The region element of the entry describes the region bounding the word or object, and the action element is `Select, W`, where `W` is the word or object, and the conventional selection action is to `copyW` to a clipboard.

The digital page uses this action table whenever it receives double clicks. [It is assumed that the digital page can distinguish double clicks as such. This may require sending a time-stamp with the click event, as transmission times from the pointer to the digital page may not be uniform. Or the pointer could recognise the click type.]

5 Suppose now we add another selection method, triple clicking to select a line. Clearly we can add an action table with an entry for each line.

Finally, let's consider another common text selection method: pushing the button down, dragging the pointer, and releasing the button elsewhere. The digital page receives the position of the pointer at the two locations.

10 This selection method is not efficiently amenable to an action-table implementation (without a separate representation of content and a mapping from positions into that representation). An implementation analogous to the ones just described would require entries for all pairs of characters.

15 We turn to the example to illustrate the concept of selecting content. The example, in which Ariane wants to have the word "exsufflicate" translated into French, is as follows:

She clicks on the box TRANSLATIONS on the top of the physical page. A special menu appears on the screen, giving a choice of several languages. She selects French. She now has a choice between either a word-in-context automatic translation or a number of standard literary translations. She selects Gide's translation, then
20 directs the pointer to the difficult word. Gide's translation of the line appears on the screen, with the word(s) corresponding to "exsufflicate" singled out in boldface.

Note that the entire line is translated, though the user pointed to a single word.

25 This can be done with action tables as follows: The TRANSLATIONS box has an entry in the active action table, or its region is type Action; either way the action puts up the menus and collects the user's decisions. The digital page has access to Gide's translation, which is aligned sentence-for-sentence, and within each sentence, word for word, with Ariane's copy of Othello. Given a sentence number and the number of a word within the sentence, the translation function finds the
30 corresponding sentence and word (or words) in the translated text. It boldfaces the word and returns the translation.

How is it given the sentence and word numbers? For the text identification, you build an action table specifically to support this function. It has an entry per word, containing the sentence number and word number.

6.6 Isomorphism

Coded substrate technology as described herein allows a person to interact with paper in much the same way as one could interact with a computer screen.

It has been described elsewhere how the position of the pointer on the page can be delivered to the digital page. This positional data is analogous to the position of the mouse on the screen, which is the information that a standard computer application receives.

There are ways in which the mechanisms for implementing coded substrate technology actions differ from the mechanisms one would use for standard computer applications (notably due to the fact that the paper does not change as a screen does).

There are two kinds of mechanisms that could be used to implement actions such as Ariane's as described herein. In one, the complete image and content of the printed page is represented, as such, in the digital page, while in the other, only the actions are represented in the digital page. The latter is functionally weaker than the former, because its actions can only use content that has been associated with the action in advance. Arbitrary text selection is a case in point; it requires a representation of the content in the digital page.

The former is referred to herein as *isomorphism* between the physical and digital page. This is illustrated in Figs 14 and 15.

It is common practice to map from a position on the screen to a position in a digital document that has been rendered on the screen; this is what is done by browsers and text editors, for example.

To make use of a representation of the content of a printed page, it must be possible to map from a position on the page to a position in the digital representation.

This can be done by using a digital representation that is isomorphic to the printed page. In this way, positions on the page match one-for-one positions on the screen.

One can imagine the printed page reproduced on a screen. As the pointer is moved on the physical page, the cursor moves, simultaneously (modulo network

Substitute Sheet (Rule 26)

delays) and in parallel. Button clicks on the pointer are treated as clicks on the mouse. The digital page process acts just as it would if the user were moving the cursor, and clicking, with the mouse.

Given this kind of parallelism, standard computer application techniques (such as those used in editors and browsers) may be used to implement Ariane's actions. Provision of support for those actions can be made using techniques known in the art.

This parallelism can be achieved by associating the digital page process with a special purpose window manager. This window manager manages the window displaying the page. The window size is identical to the page size. The window manager takes positional information from the pointer rather than the mouse---but this is transparent to the digital page process.

There is no need for this window to be rendered to a screen---it can be entirely virtual.

The isomorphism is between the image on the physical page and one aspect of the digital page: a digital representation of that image (or information from which such a representation can be constructed). The digital page contains more information, including the mapping from the image to the content and the actions to be performed.

Constructing a digital representation that looks like what you get when you print to paper was a difficult problem---most text editors and browsers display a document that differs on-screen to one degree or another from the printed version. Adobe PDF format provides page-description functionalities that are strong enough that the printed and digital representations truly look alike. Other page description formats may provide similarly.

A special-purpose window manager is just one way among many to take advantage of such an isomorphism. [Another way would be to use a standard window manager, but to modify the positional and event information as it comes from the pointer so that it appears to the window manager to be a mouse event. This moves some of the special-purpose processing, and some elements of window-management, into the communication infrastructure, which would communicate with the window manager. Alternatively, positions can be interpreted directly by the digital page process. Window management per se (which window is on top, which application handles an event) isn't relevant.

Substitute Sheet (Rule 26)

But an event handler, which is typically provided by the window manager, would have to be incorporated into the digital page.]

7. Applications

7.1 Writer-pointer.

Here, using the pointer disclosed in Fig. 11 which writes on the coded substrates, the position of the point is monitored during hand-writing through the usual localisation device. This permits the marks to be registered in digital form simultaneously to their physical production by the writer, in a transparent way. This provides the following applications:

- drawing on a coded substrate and getting a digital copy of the drawing, on which various operations can be later performed (smoothing, shape-recognition, etc.) As the pen movements are recorded in time, the digital drawing is dynamic, as opposed to the case of scanning the final state of a drawing. This dynamicity might help a shape-recognition program identify a circle, a square, etc. There are also applications in signature authentication, by making the dynamics of the strokes a part of the signature.

- taking hand-written notes during a meeting, which are later accessible in digital form; dynamicity may help perform hand-writing recognition, if an ascii representation of the text is desired.

- writing (or drawing) on a flip-chart, and keeping a digital copy; or writing with a marker on a white-board having a coded substrate, and keeping a digital copy.

- filling multi-choice paper forms.

- putting by hand editor's marks on a draft typescript and having these marks transmitted and interpreted on the digital side in order to produce a corrected version.

7.2 Controlling access rights I

As has been discussed, coded substrates have a number of applications to interacting with services. As will be seen, it also has significant potentialities for the control of access rights to data as well as to digital commerce.

Substitute Sheet (Rule 26)

Consider the representative case of a classical recording publishing company, Digital Grammophon, or DG; this company distributes musical recordings over the net, and it must find ways to have listeners pay for them. coded substrate can help to solve this problem.

5 Consider the following scenario. DG sells, through record or book shops, "listening cards" printed on coded substrate. One such card, for instance, lists the last six Mozart symphonies. Each movement is listed as a subheading under the corresponding symphony heading. The customer buys the card, goes back home, clicks on the heading for the 40th, and the music is played for him on his hi-fi
10 peripheral device.

Is this scheme commercially secure for the DG company with respect to an unscrupulous user intent on defeating copyrights?

- Such a user might legally buy a listening card for the symphonies and reproduce, through standard photocopying, its printed content on thousands of cards,
15 and distribute these through parallel channels. Of course, this scheme will not work. A buyer of such a card will not be able to use it in any way. It is instructive to compare this situation with the case where the cards produced by DG would be printed on conventional paper, and where marks printed on these cards, such as barcodes, would encode the addresses of the music tracks. In this case, a standard
20 photocopy of the original card would be functionally indistinguishable from this original card.

- Such a user might become more sophisticated and produce perfect copies of the original card, including the coded substrate (an operation, which, by the way, may be illegal if coded substrates were a monopoly of a coded substrate
25 supplier).

Although such an operation would be technically difficult and costly, it might not be impossible. However, DG can easily prevent such a problem. Each piece of coded substrate has a unique identification number page-id, which is associated at the DG site with the digital representation of the page (this is of course a many-to-one
30 mapping). Because of this unique identification for each physical card, DG can detect two simultaneous access attempts through the same page-id, and knows that this event corresponds to an illegal situation. It can then take measures of different degrees of severity. A drastic measure would be to void the guilty page-id of any future access

Substitute Sheet (Rule 26)

rights. Under such a scheme, people buying the illegal copies would be quickly discouraged.

[Other less severe schemes could be implemented, such as putting limits on how many cumulative hours a given page-id could be used each day, or on the total number of times it could be used in its lifetime. These schemes would obviously have broader commercial implications than just preventing piracy: for example, one could buy the right to listen 10 times to the "Jupiter" symphony.]

7.3 Controlling access rights II

But where the user is not to be discouraged by the measures in section 7.2 above, he make a further attempt to cheat the DG company.

Suppose the user has noticed that all the DG cards for the Mozart symphonies bear consecutive page-id numbers. From buying one such card, she can therefore predict what the thousands DG cards for these recordings look like. She then goes on and produces cards that have the right page-id numbers on them.

Now this move is potentially dangerous for DG. How can it distinguish a card bought legally from a card bought from the unscrupulous user? There is no obvious way.

But there is a radical counter-measure that DG, with support from the coded substrate supplier, can take. The coded substrate supplier, rather than simply encoding a page-id number in the white CODED SUBSTRATE sheets it sells to DG, now pairs the page-id with a hash-code page-id-code. The function mapping a page-id into the page-id-code is a secret property of the coded substrate supplier; it may be implemented in a variety of ways: through a random number generator, a secret hash-coding algorithm, or a public-key-cryptography scheme. Although there are some differences between these various approaches, what is crucial for the current problem is the following thing: the page-id-code cannot be predicted from the page-id by a person outside of the CODED SUBSTRATE SUPPLIER. On the other hand, the coded substrate supplier, when selling coded substrates sheets to DG, provides DG with a list of <page-id,page-id-code> pairs, so that DG knows the code associated with any sheet of coded substrates it owns.

Now the unscrupulous user, even if he is able to predict which page-id numbers will be used for the Mozart symphonies, has no way to know the page-id-

Substitute Sheet (Rule 26)

code numbers associated with these pages. The only way she could know these numbers would be by buying all the DG cards corresponding to the symphonies, a somewhat uncongenial prospect.

We thus see that, while coded substrates may still be copiable, given the
5 aforementioned additional technology, with a page-identifier coding scheme, it becomes unforgeable: in order to produce a usable card, the pirate needs to own an original for that card.

It should be noted here that the idea of privately encoding the page
10 identifiers does not strictly require the coded substrate medium. It could also be implemented via conventional printed marks on the card. Although this might already be an interesting possibility in itself, the coded substrate medium fits the bill very well here, because one generic architecture (paper, pointer, net infrastructure), already employed in several other functions, can be readily adapted to the case at hand. The only modification of the original scheme consists in adding a page-id-code to the
15 page-id during the production phase at the coded substrate supplier, an operation with minimal supplementary cost.

Another access rights example is as follows. The user buys stamp at local news-stand, transfers its code to mail-order house, mail-order house transfers code to bank

20 Suppose the coded substrate supplier, under the supervision of a commercial bank, produces small pieces of adhesive coded substrates, shaped like post stamps, and printed in various denominations (10cts, 1\$, 10\$, etc.). These stamps are sold, under a sealed wrapper, at the bank offices, through local news-stands, etc. Using coded substrate technology, the stamps are provided with a unique page-id and
25 with a corresponding page-id-code (see previous section). The pair <page-id ,page-id-code > is known only to the bank.

A customer wishes to purchase some catalogue item through a mail-order company. She buys enough stamps to cover her order, sticks them on an empty space provided next to the item in the catalogue, and then, using the pointer, clicks first on
30 the item, then on each of the stamps. The action of clicking on a stamp has the effect of transferring the <page-id ,page-id-code > pair to the mail-order company. The mail-order company can then transfer these codes to the emitting bank, which credits the mail-order company's account for an amount equal to the stamp denomination.

Substitute Sheet (Rule 26)

The bank will accept an operation with a given <page-id ,page-id-code > pair only once: any attempt to re-use this pair for a later operation will be rejected, so that each stamp can effectively be used for payment only once in its lifetime. After valid payment has been confirmed by the bank to the mail-order company, the item is sent to the customer.

The procedure just sketched needs to be compared with other forms of digital payment. Still, noteworthy is the fact that, using simple coded substrate technology, a form of commercial transaction – which preserves the anonymity of the buyer of stamps – can be provided relatively easily.

Substitute Sheet (Rule 26)

CLAIMS

1. A method for producing identifiable sheets comprising:

5 (A) obtaining sheet medium; and

(B) producing one or more identifiable sheets from the sheet medium;

in which (B) comprises:

10

(B1) printing machine-readable markings within one or more sheet areas of the sheet medium, each sheet area being an area of the sheet medium from which one of the identifiable sheets is produced; the markings within each sheet area defining two or more zones of the sheet area; the markings within the zones of each sheet area
15 indicating a sheet identifier; the sheet identifier identifying the identifiable sheet produced from the sheet area.

2. The method of claim 1 in which the markings within each zone of each sheet area further indicate a position of the zone within the sheet area.

20

3. The method of claim 2 in which (B1) comprises printing, within each zone, sheet identifier markings indicating the sheet identifier and position markings indicating the position.

25

4. The method of claim 1 in which (B1) comprises printing the machine-readable markings with an invisible ink.

5. An identifiable sheet comprising:

30

an area of sheet medium; and

machine-readable markings within the area of sheet medium; the markings within the area of sheet medium defining two or more zones of the area of sheet medium; the

Substitute Sheet (Rule 26)

markings within the zones of the area of sheet medium indicating a sheet identifier;
the sheet identifier identifying the identifiable sheet.

5 6. The identifiable sheet of claim 5 in which the markings within each zone of
the sheet area further indicate a position of the zone within the sheet area.

10 7. The identifiable sheet of claim 6 in which the markings within each zone
include sheet identifier markings indicating the sheet identifier and position markings
indicating the position.

8. The identifiable sheet of claim 5 in which the machine-readable markings are
invisible.

Substitute Sheet (Rule 26)

1/11

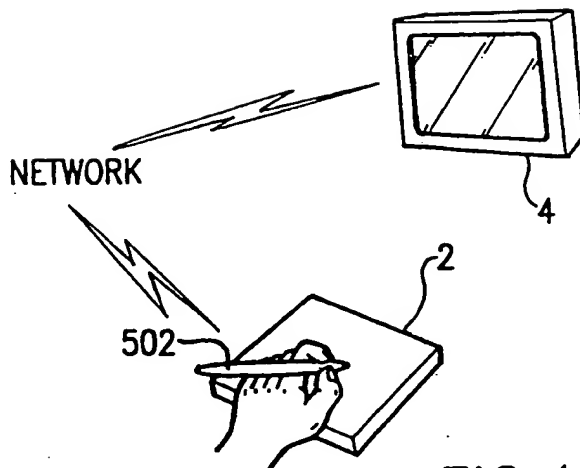


FIG. 1

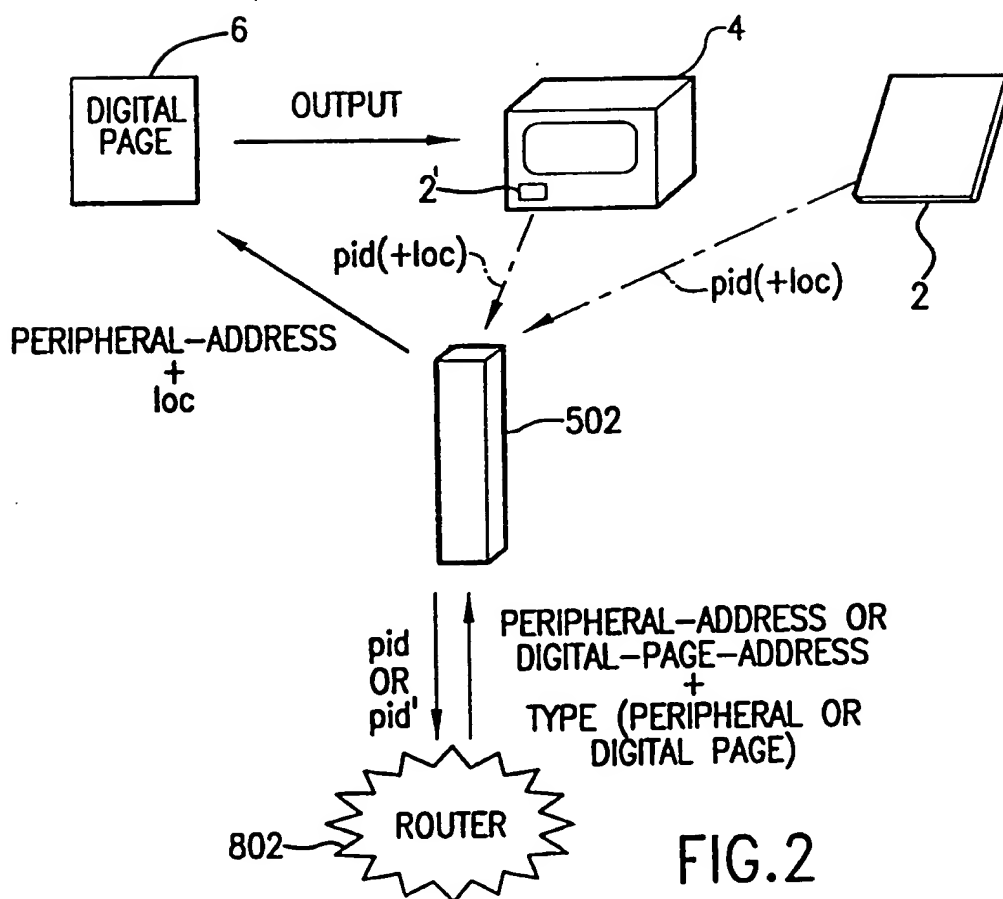


FIG. 2

Substitute Sheet (Rule 26)

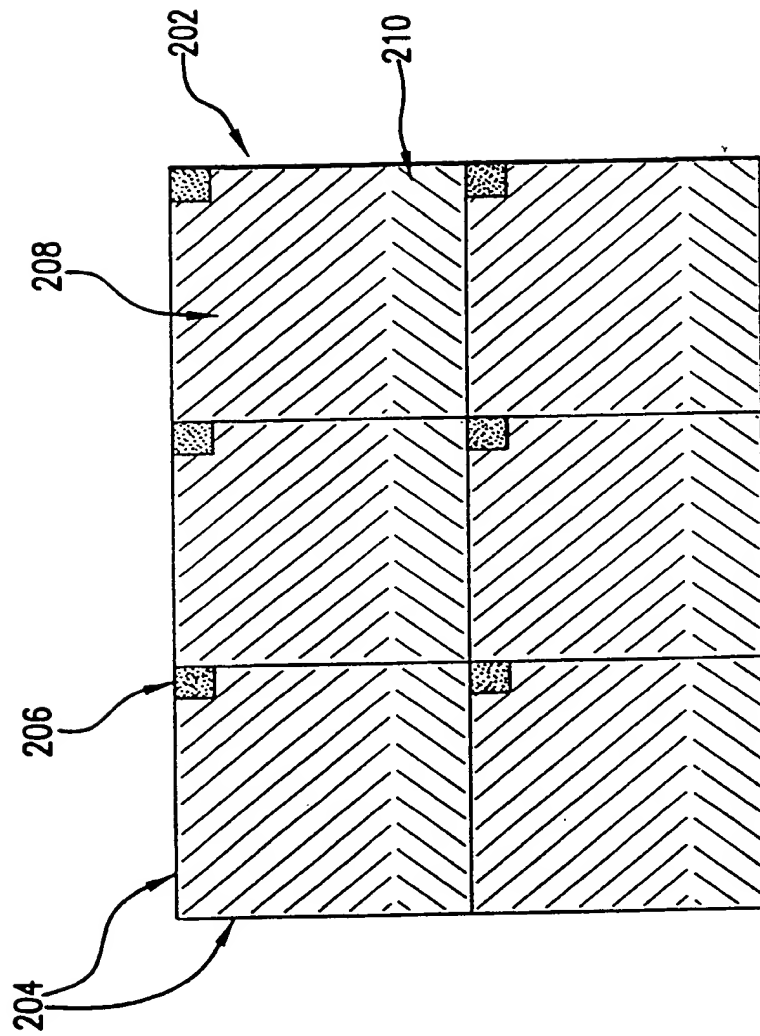
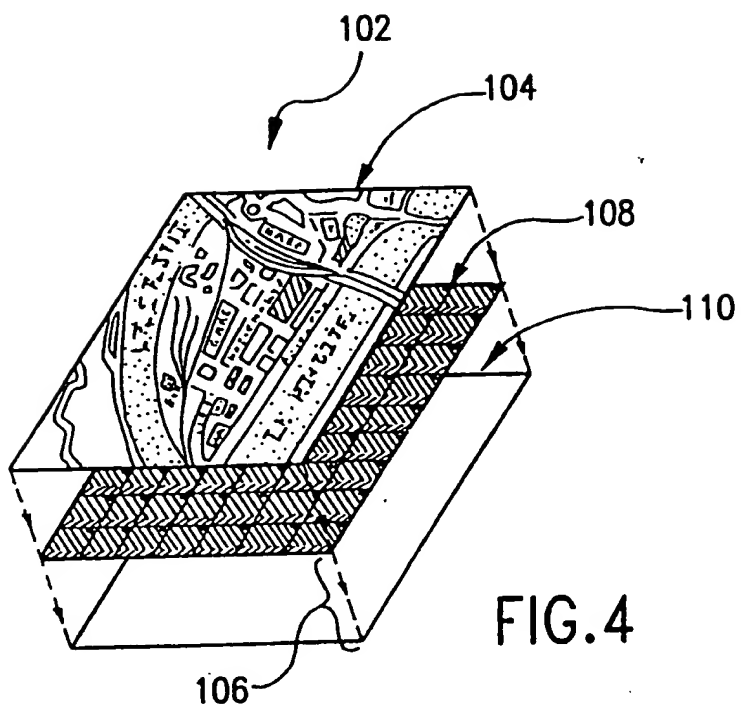


FIG. 3

3/11



Substitute Sheet (Rule 26)

4/11

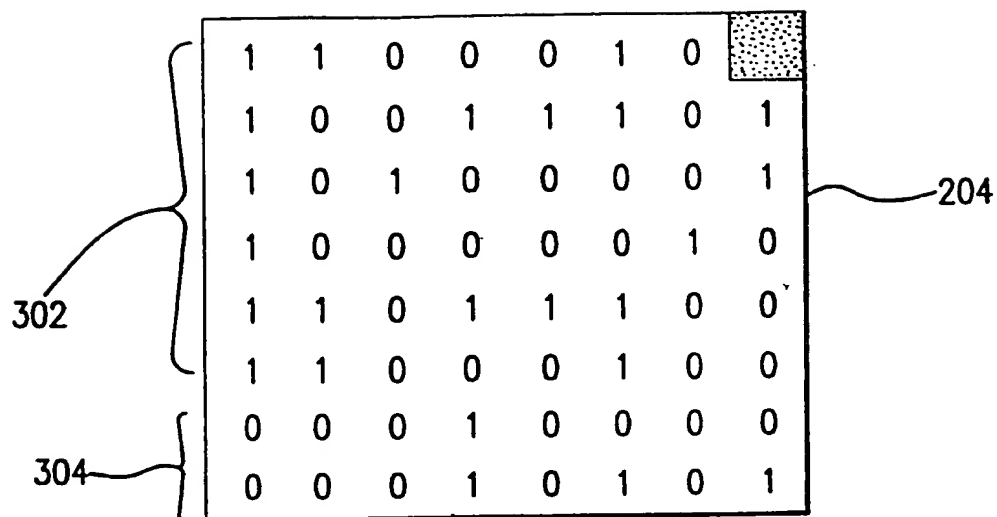


FIG. 5A

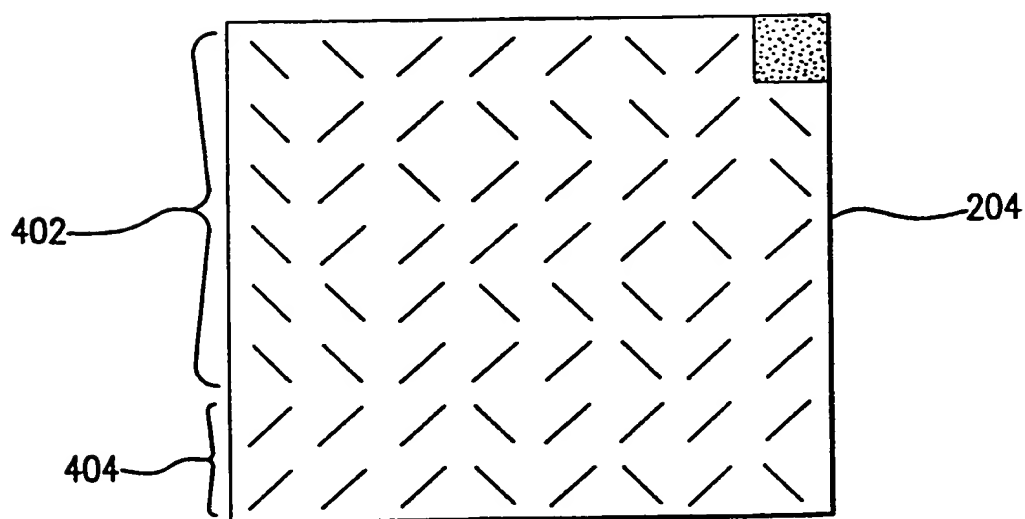


FIG. 5B

Substitute Sheet (Rule 26)

5/11

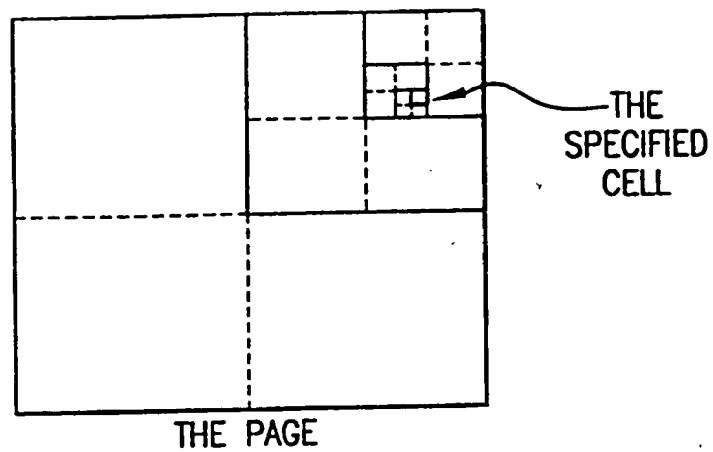


FIG.6A

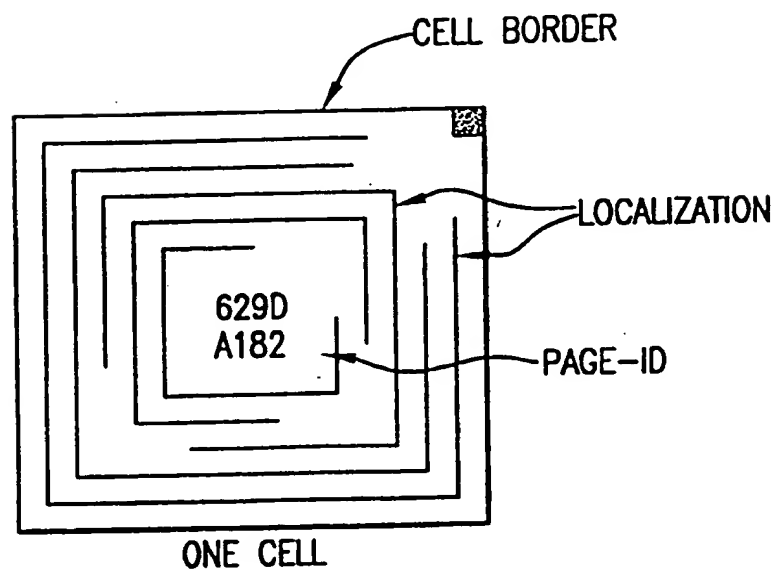


FIG.6B

Substitute Sheet (Rule 26)

6/11

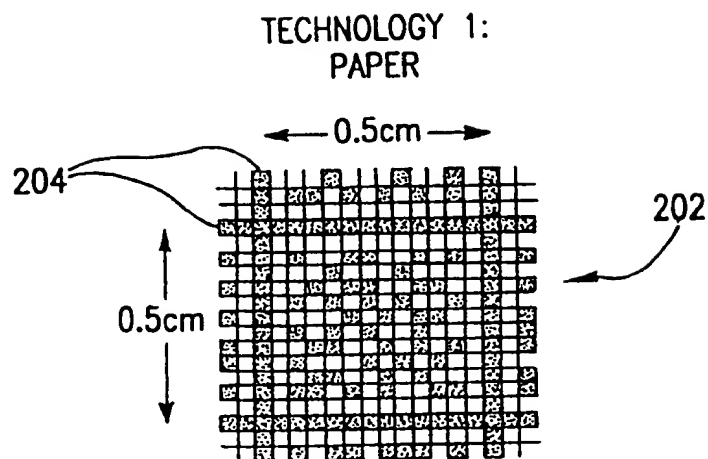


FIG. 7

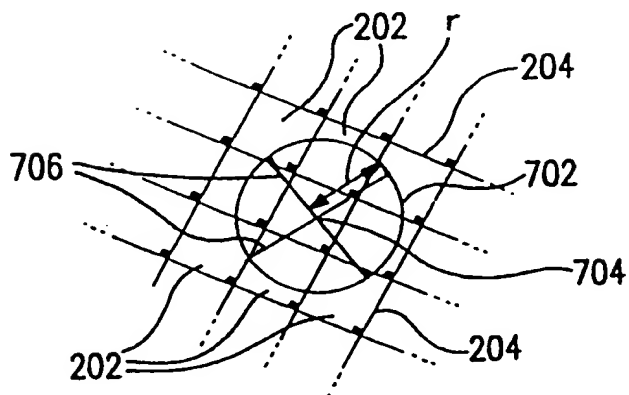


FIG. 10

Substitute Sheet (Rule 26)

7/11

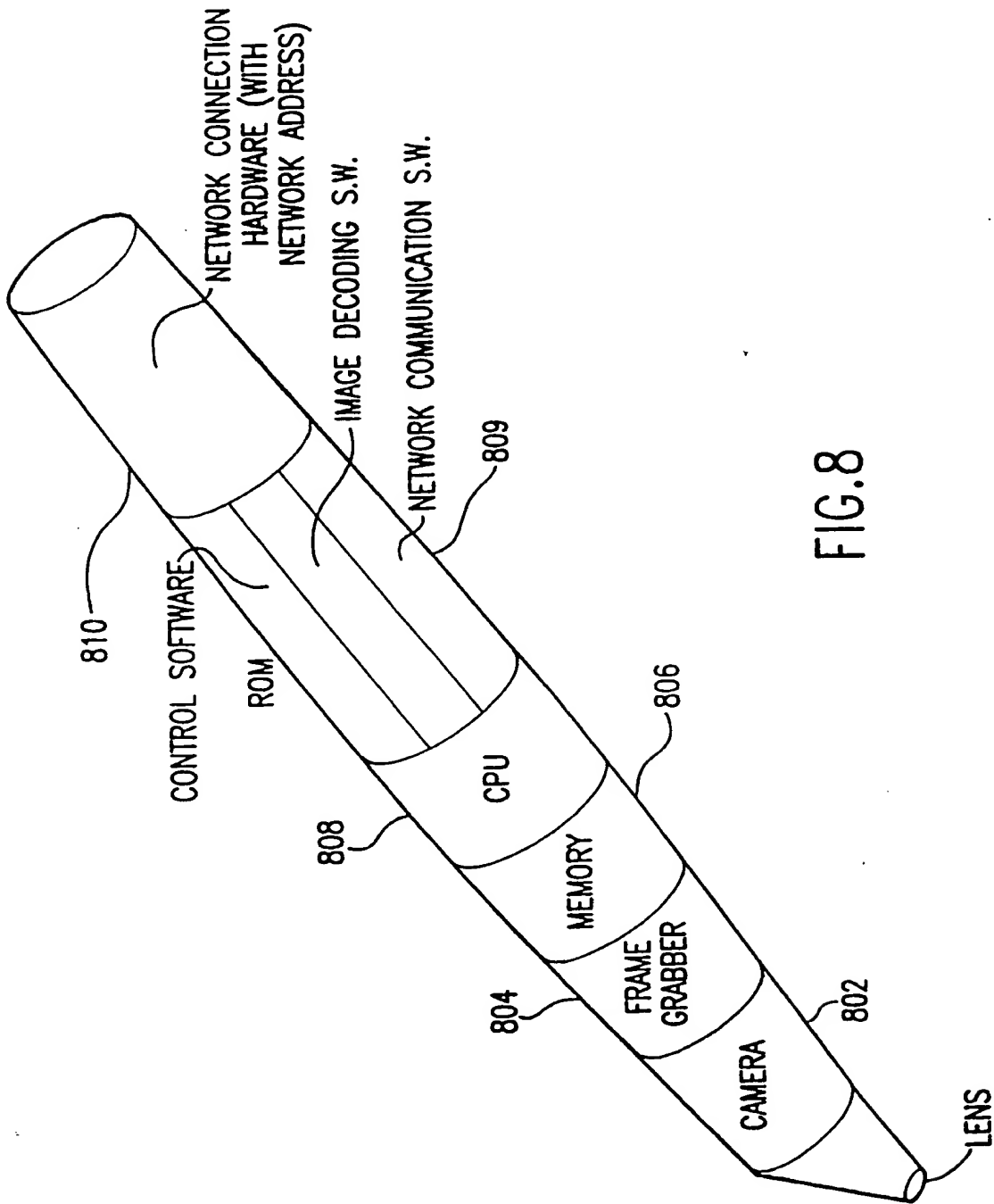
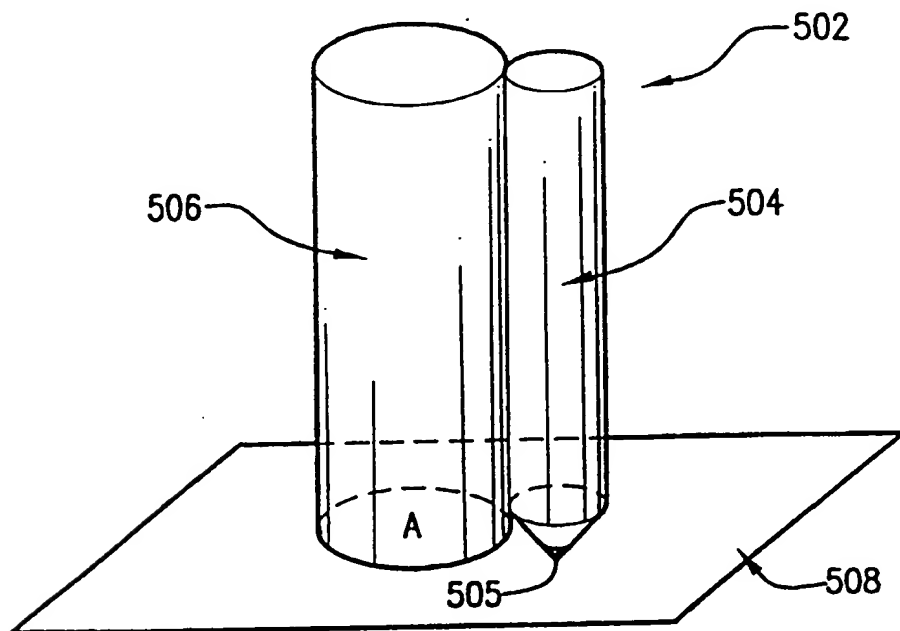
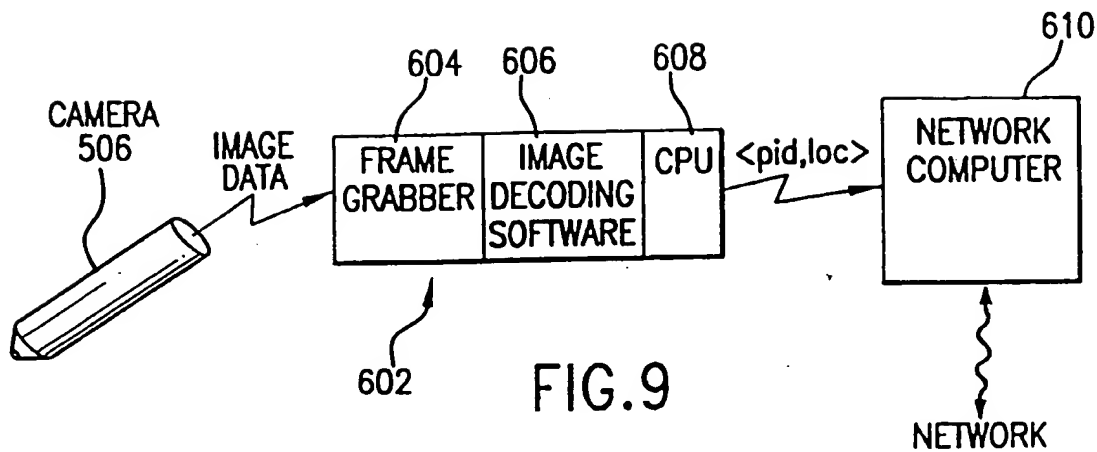


FIG. 8

Substitute Sheet (Rule 26)

8/11



Substitute Sheet (Rule 26)

9/11

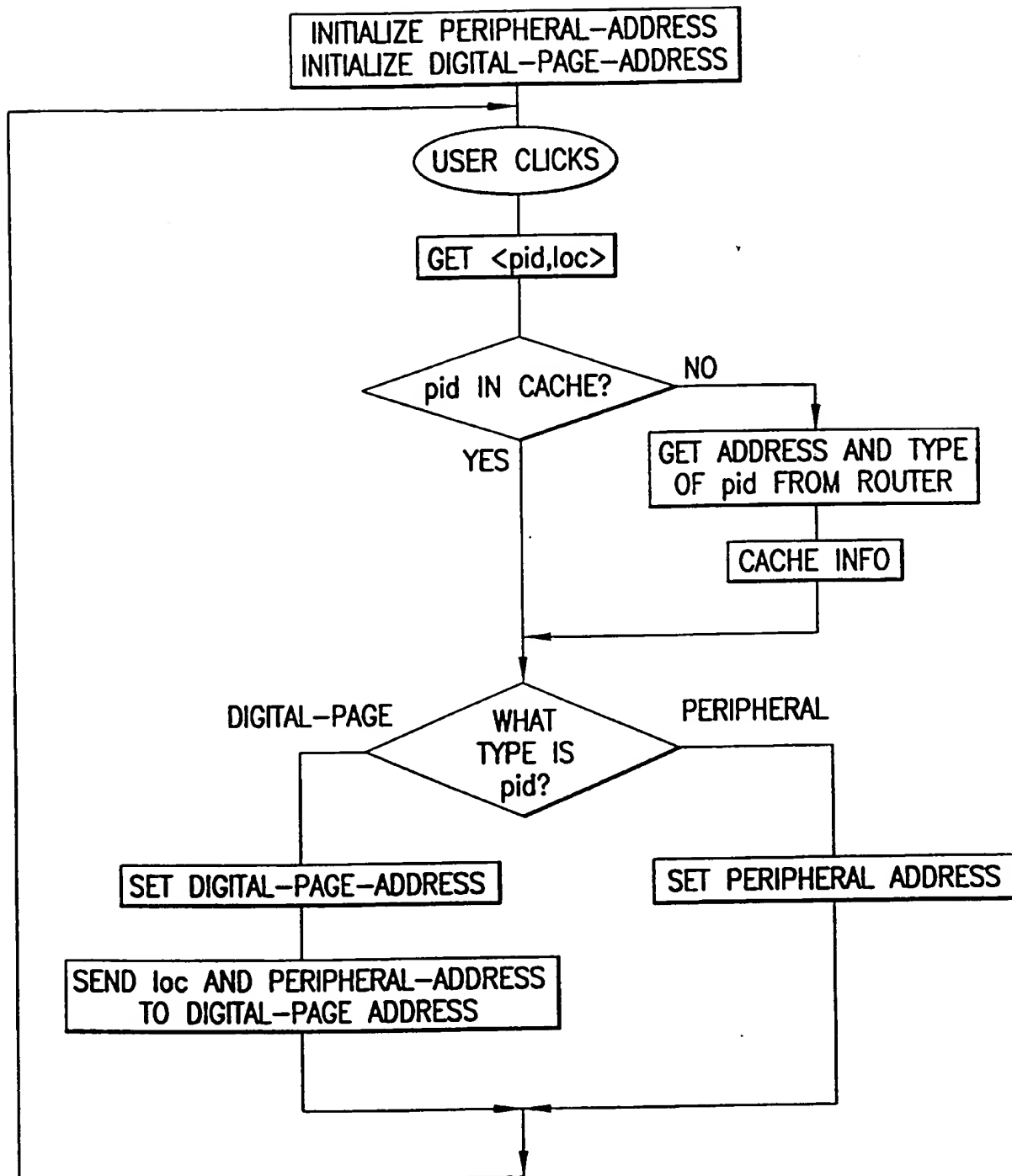


FIG.12

Substitute Sheet (Rule 26)

10/11

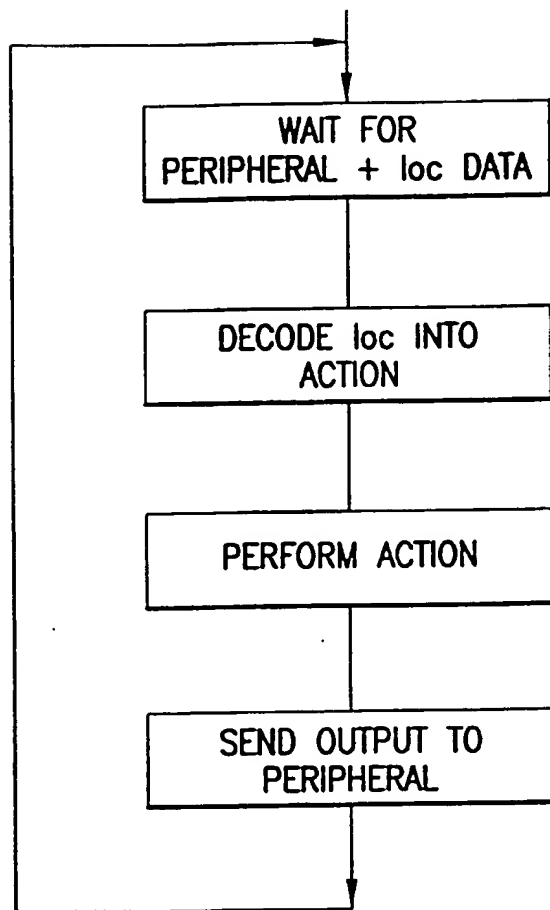


FIG.13

11/11

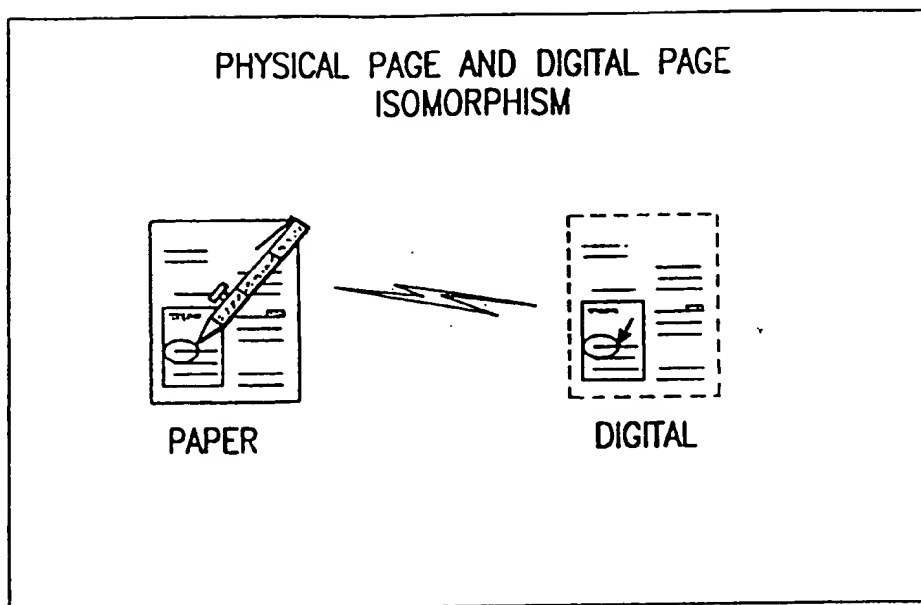


FIG.14

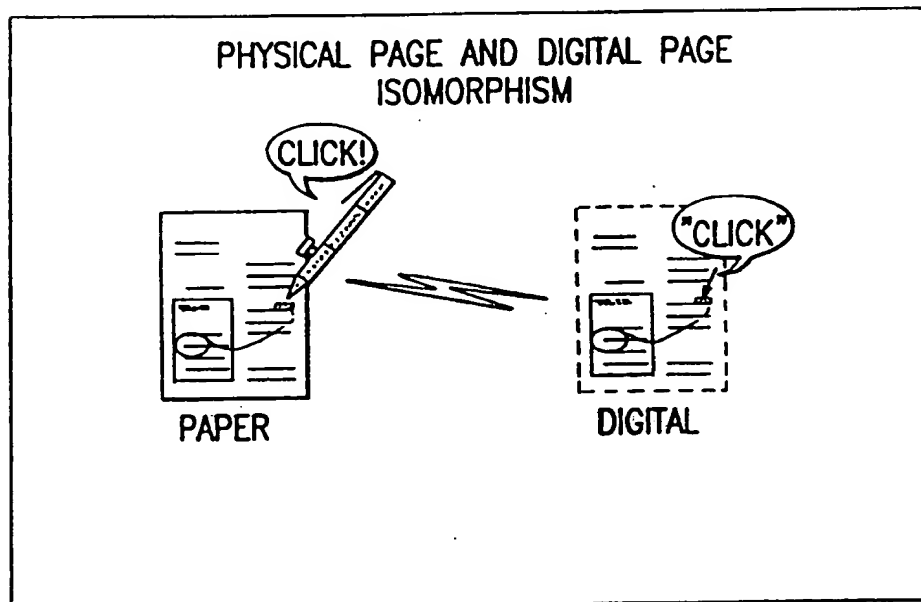


FIG.15

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/20597

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06K 19/06
US CL :235/494

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 235/494, 487

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
none

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

U.S. PTO APS

search terms: glyph?, region, zone, print?, code, infrared ink

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,128,525 A (STEARNS et al.) 07 July 1992 (07/07/92), col 5 line 42 - col 6 line 39.	1-3,5-7
Y	US 5,221,833 A (HECHT) 22 June 1993 (22/06/93), see column 3 lines 10-47.	1-3,5-7
Y	US 5,311,017 A (FILO) 10 May 1994 (10/05/94), see column 7 lines 8-9.	4,8
A	US 5,486,686 A (ZDYBEL, JR. et al.) 23 January 1996 (23/01/96), see entire document.	1-3,5-7
A	US 5,449,896 A (HECHT et al.) 12 September 1995 (12/09/95), see entire document.	1-3,5-7
A	US 5,453,605 A (HECHT et al.) 26 September 1995 (26/09/95), see entire document.	1-3,5-7

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*A* document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means	
P document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

29 DECEMBER 1998

Date of mailing of the international search report

09 FEB 1999

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

KARL D. FRECH *for*
Telephone No. (703) 305-3491

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/20597

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,449,895 A (HECHT et al.) 12 September 1995 (12/09/95), see entire document.	1-3,5-7

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*